

TEKNILLINEN KORKEAKOULU  
Tietoliikenneohjelmistojen ja Multimedian Laboratorio  
T-111.080 Sisällöntuotannon seminaari  
Syksy 2003: Näkökulmia vuorovaikutteisuuteen

3.11.2003

# **Pelien kokeellinen modaus**

Ville-Kalle Arponen

47915U

# Pelien kokeellinen modaus

Ville-Kalle Arponen  
TKK, Tietoliikenneohjelmistojen ja Multimedian Laboratorio

ville-kalle.arponen@hut.fi

## Tiivistelmä

*Paperi käy läpi pelejä modifioimalla saavutettuja tieteellisiä päämääriä ja projekteja. Projekteja katsotaan myös tekniseltä kannalta keskittyen pelien modulaarisen, vapaan ohjelmakoodin tarjoamiin mahdollisuuksiin. Lopuksi pohditaan pelien modaamisen tarjoamia käyttämättömiä mahdollisuuksia.*

## 1 PELIEN MODIFIOINTI ELI MODAUS

Pelin modaus on pelin muuttamista uudeksi peliksi. Pohjana on aina olemassa oleva peli jonka komponentteja lähdetään muuttamaan ja jopa lisäämään. Modaamisella saavutettu uusi peli saattaa olla yllättävän erilainen alkuperäiseen nähden, mutta koska pelin grafiikkamoottori ei useinkaan ole modattavissa on uuden pelin jossain määrin toimittava tuon moottorin asettamissa rajoissa.

Modaamista on monentasoista ja vaikka syvimmän tason modaaminen aina vaatiikin ohjelmakoodiin puuttumista, joissain peleissä on valmiina editoreita joilla modaamista voi harrastaa ilman suurempaa ohjelmointitaitoa. Yksinkertaisimmillaan pelejä modataan tekemällä niihin uusia kenttiä eli muokkaamalla pelin ympäristöä. Vaikeimmillaan puututaan ohjelmakoodiin ja tehdään siihen lisäyksiä.

Pelejen modaaminen on yleisintä nimenomaan pelaajien keskuudessa. Pelaajien tekemät modit eli modifioidut pelit ovat usein vitsailumielessä väännettyjä, mutta osaa modeista pelaa jo mielellään. Yksi tunnetuimmista modeista on Half-Life pelin modi Counter-Strike, joka on lähes yhtä suosittu kuin alkuperäinen peli (Bezold & Diekman, 2003, s.34-41).

Pelaajien innostus modaukseen on huomattu myös pelitaloissa. Esimerkiksi Neverwinter Nightsin pelisisältö oli varsin vähäpätöinen, kun se julkaistiin. Pelin fokus oli kuitenkin hyvien modausmahdollisuuksien tarjoaminen pelaajille, mikä oli onnistunut strategia. Nyt noin vuosi pelin ilmestymisen jälkeen, on Internetissä tarjolla jo lähes 3000 Neverwinter Nights modia (<http://nwn.bioware.com>, 2003).

## 2 PELIEN MODIFIOINTI TIETEELLISIIN TARKOITUKSIIN

Voimakkaat pelimoottorit, pelien ohjelmoinnin modulaarinen rakenne ja vapaa lähdekoodi ovat kiinnostavia tekijöitä myös tieteen kannalta. Pelien avulla voi kokeilla ja on kokeiltu uusia konsepteja ja edistetty olemassa olevia.

### 2.1 Syyt pelien käyttöön

Quake on usein tieteen piiristä lähtöisin olevan modauksen kohde. Quake sisältää voimakkaan grafiikkamoottorin, joka renderöi reaaliaikaisesti kohtuullisen kokoista 3d maailmaa, on testattu, mutta ennenkaikkea on peli.

Jotta pelin saa myytyä pelaajille on pelin hinnan oltava kohdallaan. Quake 3, Quake sarjan uusin versio, maksaa noin kolmekymmentä euroa ([www.dose.fi](http://www.dose.fi), 2003), ja se toimii normaalihintaisella uudehkolla multimediatietokoneella. Vastaavan suorituskyvyn hankinta perinteisillä tietokonegrafiikan ohjelmistoilla ja laitteistoilla tulisi selkeästi kalliimmaksi.

Pelien hyviin puoliin hinnan lisäksi kuuluu huikea suorituskyky. Pelimoottorit on hiottu ottamaan irti mahdollisimman paljon kuluttajien laitteistoista. Osa suorituskyvystä on saavutettu ottamalla vapauksia fysiikan ja inhorealismien suhteen, mutta mikäli tarkoituksena on saavuttaa uppoutuminen eikä tarkka mallinnus, ovat nuo oikotiet hyväksyttävää.

Hintalaatusuhteen korkeus on ymmärrettävää. Viihdeteollisuuden ympärillä pyörii suuret määrät rahaa ja siten myös resursseja. Peliteollisuudessa sattuu vielä lisäksi tällä hetkellä liikkumaan enemmän rahaa kuin missään muussa viihdeteollisuuden haarassa.

Jotta syitä pohdittaessa pysyttäisiin totuudessa niin on syytä nostaa pöydälle tietotekniikan tutkijoiden kiinnostus peleihin. Osa tietotekniikan tutkijoista on alun perin päätenyt tietotekniikka-alalle juuri pelien johdattamana, joten on luonnollista että tutkijat ovat kiinnostuneet peleistä. Modauksella saavutetuista tieteellisistä sovelluksista suuri osa on nimenomaan peittelemättömästi pelejä.

### 2.2 Tieteellisen modauksen päämäärät

Mitä peleillä voi tehdä muuta kuin pelata niin kuin on perinteisesti ollut tapana?

Kysymykseen on jo keksitty monta vastausta. Asetetaan pelimaailman elementeille ja tapahtumille todellinen relaatio todellisen maailman tapahtumiin ja olioihin. Käytetään voimakasta pelimoottoria hyväksi perinteisten virtuaaliympäristöjen renderöinnissä. Otetaan pelistä hirviöt ja lisätään ne oikeaan maailmaan. Simuloidaan todelliseen maailmaan tarkoitettujen innovaatioiden toimintaa pelimaailmassa. Avataan tekoälyille temmellyskenttä tai tutkitaan tekoälyä.

Lisäksi on suuri joukko modeja, jotka on modifioitu vain pinnallisesti kuten Fuchsin ja Eckermanin taidekokeilut (Fuchs & Eckerman, 2002/2003). Keskityn tässä kuitenkin

sovelluksiin, joissa pelejä on modattu pintaa syvemmältä ja saavutettu siten ennennäkemättömiä teknisiä saavutuksia.

### *2.2.1 KYBERAVARUUS JA PSDOOM*

William Gibsonin Kyberpunk-maailmoista tuttu, 1981 Vernon Vingen kehittämä, idea kyberavaruudesta (Gibson, 1984 /Vinge, 1981) varmasti kiehtoo jokaista tieteiskirjallisuuden ystävää. Kyberavaruus on ikään kuin 80-luvun kuvitelma siitä miten internet toimii.

Kyberavaruus ei kuitenkaan ole kuin internet sillä kyberavaruudessa liikutaan kuin pelissä, avatarin avulla. Dekkeri, henkilö joka osaa liikkua kyberavaruudessa, käyttää todellisen maailman esineiden kopioita myös interaktiossaan kybermaailmassa. Tärkeän datalinnoituksen ympärille voidaan rakentaa ”fyysinen” muuri, ja koiria partioimaan ympärille. Murtautumishaluinen dekkeri saattaa omistaa tulipallo-ohjelman jolla koirat voi polttaa poroksi ja poran jolla voi murtautua muurista läpi. Datalinnoitus voi lisäksi sisältää esimerkiksi kontrollit datalinnoituksen omistavan yrityksen toimitilojen hisseihin. Olennaista on kuitenkin se, että datalinnoitus johon murtaudutaan näin pelimäisesti sisältää todellista tietoa.

Kyberavaruudessa tehtävien tempausten vaikeus ja niihin vaadittava työmäärä korreloi niiden aiheuttamiin efekteihin todellisessa maailmassa.

PSDoom on kokeilu kyberavaruustyylisen ideologian toteuttamiseksi (Chao, 2001).

PSDoom itsessään on ID Softwaren kehittämän Doomien modi. Pelistä on modattu hirviöiden relaatio todelliseen maailmaan. Yksi hirviö pelissä tarkoittaa yhtä prosessia tietokoneessa. Surmaamalla hirviön voi tappaa prosessin. Hirviöt saattavat alkaa tapella keskenään, kuten alkuperäisessä pelissäkin, jolloin on mahdollista että prosessit tappavat toisiaan ilman pelaajan / ylläpitäjän väliintuloa.

Periaatteellisella tasolla prosessien mielivaltainen listiminen kuulostaa varsin mielipuoliselta tavalta ylläpitää järjestelmää, mutta todellisuudessa tällainen prosessien satunnainen tappaminen on osa täysin normaalia ylikuormittuneen systeemin resurssienhallintaa. Ongelmaksi modissa muodostuu prosessien prioriteetit jolloin vihastunut tekstieditori saattaa vaikkapa kaataa koko systeemin tappamalla jonkin kriittisen prosessin. Lisäksi pelin hirviöt ovat liian agressiivisia.

Mitä hyötyä on kyberavaruuden implementoinnista? Jos aletaan tutkia prosessien käytöstä normaalissa tietokoneessa niin voidaan sanoa, että ne kisaavat keskenään koneen prosessoriajasta, muistista, prioriteeteista ja jopa siitä ettei ylläpitäjä päättäisi tappaa niitä. PSDoomin fyysisesti taistelevat prosessit antavat kyseiselle kilpailulle aivan uuden näkökulman. Mikäli PSDoomin moottoria hiottaisiin pidemmälle ja prosesseille annettaisiin tekoälyä saattaisi pelin kautta kehittyä aivan uudenlainen oikeasti hyödyllinen prosessinhallintatyökalu.

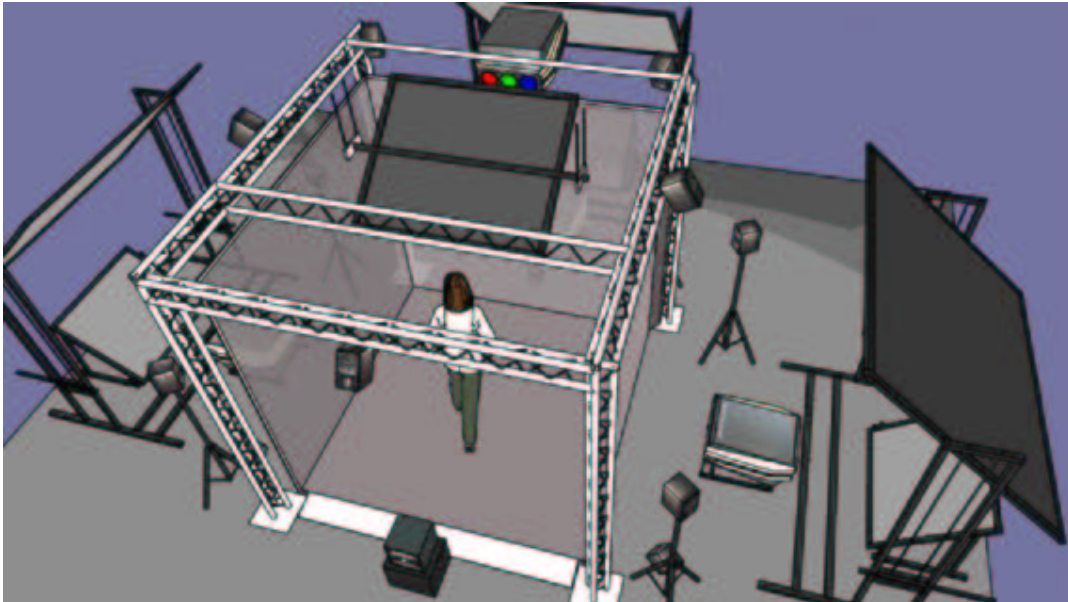
Lisäksi PSDoom nostaa esiin aiheen hyötyohjelmien välttämättömästä tylsyydestä. Doomia pelaava ylläpitäjä jaksaa varmasti olla huomattavan paljon pidempään kiinnostunut prosessiensa hyvinvoinnista kuin tylsää prosessilistausta tuijottaessaan.

### 2.2.2 VIRTUAALITODELLISUUS JA CAVEUT

Virtuaalitodellisuuslaitteiden ja –sovellusten tarkempi esittely ei kuulu tämän paperin alueeseen, mutta sanottakoon että todelliset implementaatiot ovat usein kalliita ja hankalia. Yksi yleisimmistä yliopistomaailman toteutustapoja on Teknisessä Korkeakoulussakin oleva EVE (ks. kuva 1), joka tunnetaan myös tekijänoikeuksin suojatulla nimellä CAVE.

EVE on kuutiomainen rakennelma jonka eri seinille heijastetaan 3d laseilla nähtävää kuvaa. Käyttäjän on tarkoitus seistä tai liikkua keskellä kuutiota, jolloin saavutetaan kolmiulotteinen maailma. Kuution kulmat eivät näy koska kuvat projisoidaan siten, että ne jatkuvat toisiinsa reunoilla.

EVE pyörii OpenGL komennoilla ja vaatii tavallisesti voimakkaan koneen pyörittämään heikkoakin esitystä.



*Kuva 1: Seppo Äyräväisen piirtämä EVE:n malli (TKK/TML, 2003)*

Unreal Tournament (UT) on Quaken kaltainen peli. UT:n monen pelaajan pelit vaativat palvelimen joka pitää huolta pelaajien interaktiosta sijainnista ja niin edelleen. Pelaajat kytkeytyvät palvelimeen ja voivat pelata samassa maailmassa.

CaveUT:n innovaatio on jakaa EVEN prosessointi n+1:lle koneelle, jossa n tarkoittaa kuutiosta rakennettujen seinien määrää sekä antaa Unrealin voimakkaan grafiikkamoottorin hoitaa renderöinti (Jacobson & Hwang, 2002).

Eri seinät ovat eri pelihahmojen näkymiä, joita jokaista siis renderöi oma tietokoneensa. Pelihahmot on liimattu yhteen niin että jokainen on 90 asteen kulmassa toisiinsa. Vain yksi seinistä on kuitenkin todellinen pelaajan kontrolloiman pelihahmon näkymä muiden seinien ollessa toimintakyvyttömiä katsojia. Pelihahmon liikkuttaminen kontrolloi siis kaikkia n hahmoa samanaikaisesti. Palvelin hoitaa synkronoinnin niin, että sivulla, alhaalla, ylhäällä ja jopa takana näkyvä näkymä tuntuu olevan loogisesti samasta paikasta eri suuntiin katsottuja näkymiä.

Koska Unreal Tournament antaa käskynsä OpenGL:llä, ei stereoskopisen kuvan ja EVE-kuution kulmien yhtenevän projektion lisääminen ole vaikeaa, etenkin kun molempiin löytyy valmiit rutiinit normaalin Caven käytön puolelta.

Kysymys kuuluu, miksi käyttää Unreal Tournamenttia eikä normaaleja grafiikkaohjelmia. Jälleen voi antaa vastauksia liittyen pelien kiinnostavuuteen ja haluun pelata UT:ta virtuaalimaailmassa. Mutta edellistä tärkeämmäksi nostaisin UT:n helpot keinot luoda uusia kenttiä eli virtuaalimaailmoja sekä UT:n ylivoimainen kyky näyttää näyttävää grafiikkaa. Esimerkiksi TKK:n EVE:n polvilleen saattamiseen ei tarvita montaakaan polygonia ja uuden sovelluksen ohjelmoiminen on aina haaste. CaveUT olisi vastaus näihin molempiin ongelmiin.

### 2.2.3 LISÄTTY TODELLISUUS JA ARQUAKE

Lisätty todellisuus (Augmented Reality (AR)) tarkoittaa tässä tapauksessa tietokonegrafiikan heijastamista normaalin näkymän päälle siten, että heijastetun grafiikan voisi kuvitella kuuluvan näkymään.



*Kuva2: ARQuaken pelaaja täydessä varustuksessa(Thomas et al. / University of South Australia, 2002)*

ARQuake on Quake oikeassa maailmassa (Piekarski & Thomas, 2002). Pelaaja asettaa puoliläpinäkyvän virtuaalikypärän päähänsä, ottaa rekyylillä varustetun leikkiaseen käteensä ja lähtee marssimaan ympäriinsä etsimään vihollisia. Siis kaikki tämä oikeasti. Paitsi että viholliset heijastetaan puoliläpinäkyviin lasihin jotka pelajaajalla on silmillään (ks. kuva 2).

Quake varmasti valittiin projektiin helppoutensa vuoksi. Pelimoottori oli valmiina ja mallinustyökalut saatavilla. Jotta peli saatiin toimivaksi oli pelialueen rakennukset mallinnettava jotta hirviöt eivät näkyisi läpi taloista. Talot värjättiin pelissä mustaksi jolloin ne eivät heijastu lasihin. Quake tarjosi valmiit hirviöt, tekoälyn, renderöinnin ja mahdollisuuden muokata bittivirtojen sisään ja ulostuloja.

Sen lisäksi että ARQuake kuulostaa uskomattoman hienolta kokemukselta ja tulevaisuudenvisiolta, täytyy sanoa että todellinen toimiva lisätyn todellisuuden sovellus rikkoo joitain keinotodellisuuden rajoja. Ensinnäkin käytännössä kaikki mitä pelissä tehdään tehdään kuten oikeassa maailmassa. Liikkuminen tapahtuu kuten normaalisti, päätä voi kääntää ja aseella ampua. Lisäksi resoluutio on uskomattoman hyvä, onhan kyseessä oikea maailma.

Voitaneen sanoa että tulevaisuudessa äiti ei enää huudakkaan: ”Nyt lapset loppuu tietokoneella pelaaminen; ulos leikkimään!” vaan ”menkääs lapset siitä kirjoja lukemasta vaikkapa ulos pelaamaan tietokoneella!”

#### *2.2.4 SIMULOINTI JA QUAKESIM*

Simuloinnilla tässä yhteydessä tarkoitetaan todellisen maailman simulointia jonkin asian testaamista varten.

On turha kuvitella, että pelit olisivat ratkaisu kaikkiin em. määritelmän mukaisiin simulointiongelmiin, mutta joissain tapauksissa pelimaailma saattaa olla mitä parhain. QuakeSimin tapauksessa sanottakoon että periaatteellisella tasolla mikä tahansa virtuaalimaailma olisi kelvannut, mutta jälleen kerran Quaken tarjoamat edut ovat suuret.

QuakeSimin idea oli testata kontekstiriippuvaisia palveluita. Kyseisessä tapauksessa testattiin laitetta, joka automaattisesti kertoi käyttäjälle erinäistä informaatiota riippuen tämän paikasta ja suuntauksesta. Tarkoitus oli simuloida laitteen käyttöä demonstraatiomielessä ja kehityksen aikana (Bylund & Espinoza, 2002).

Bylundin ja Espinozan mielestä tällaisen simulaattorin pitäisi kyetä tuottamaan todellisenkaltainen tila esineineen ja eläimineen, antaa mahdollisuus simulaatioon monelle pelaajalle samaan aikaan, antaa mahdollisuus tuottaa helposti uusia tiloja, esineitä ja avatareja sekä simuloida erilaisia sensoreita joiden ulosanti voidaan syöttää testattavalle palvelulle. Quake sellaisenaan kykenee kaikkeen muuhun paitsi viimeeksi mainittuun. Modattuna myös siihen.

Tällaisen simuloinnin hyödyllisyys on melko selkeä jo sellaisenaan. Projekti alunperinkin lähti tarpeesta eikä ”Teimme sen koska sen saattoi tehdä” -tyylisestä ideasta, joten seurauksena ei ole niinkään peli vaan todellakin simulaatio.

### 2.2.5 TEKOÄLY

Tekoälytutkimusta on oikeastaan varsin luonnollista harrastaa pelien ympärillä, sillä pelit ovat eräs yleisimmistä tekoälyn käyttökohteista. On hieman vaikea kuvitella miten nykuteknologialla voitaisiin tutkia ihmismäistä tekoälyä istuttamalla robotteja mukaan ihmisten normaaleihin askareihin. Robotit eivät ole siihen riittävän kehittyneitä, eikä myöskään tekoäly.

#### 2.2.5.1 Soar

Ihmismäisen tekoälyn tutkimukseen tarvitaan jollain tavalla yksinkertaistettu ympäristö ja ihmisten yhteisö, jossa kaikki tapahtuu määriteltyjen sääntöjen mukaan. Lisäksi vaaditaan normaalin maailman omaisesti nopeita päätöksiä ja interaktiota.

Soar projektissa Quaken pelihahmoja on annettu tekoälylle kontrolloitavaksi ja pyritty tekemään niistä mahdollisimman ihmisen kaltaisia. Koska perus Quake rajoittuu varsin pitkälti silkkään räiskintään, ei se sellaisenaan riittänyt Soarin kehittäjille kovin pitkään vaan pelimaailman lainalaisuuksia mietittiin uudelleen ja onnistuttiin saamaan aikaan peli jossa on tarkoituksena mm. säilyttämällä saada tavoitteet suoritettua. (Laird, 2002)

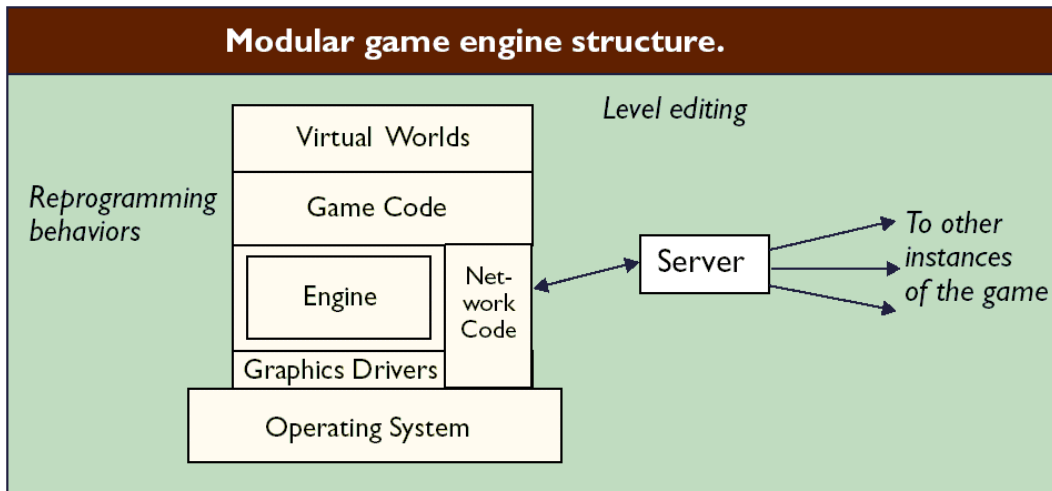
#### 2.2.5.2 GameBots

GameBots on Unreal Tournamentin modi joka mahdollistaa ulkoisten tekoälyjen liittämisen Unreal Tournament peleihin. GameBots tarjoaa liitännässään ihmismäiset havainnot, jolloin GameBotsiin kytkettävä tekoäly ei pääse ”huijaamaan” käyttämällä suoraan pelin sisäisiä muuttujia (Gaminka et al., 2002)

GameBots on erityisen mielenkiintoinen tilanteissa joissa peliä pelaavat samanaikaisesti tekoälyt ja ihmiset.

## 3 TIETOKONEPELIEN MODAUKSELLE TÄRKEÄT OMINAISUUDET JA KOMPONENTIT

Miksi pelit sitten ovat niin otollisia modifioinnin kohteita? Edellisissä osioissa on jo käsitelty pelien ylivertaisia grafiikkamoottoreita ja yleensä pelimaailmoja uudenlaisina tapoina sovelta tieteellisiä ideoita. Pelien soveltuvuuteen on kuitenkin myös projektin toteuttamista helpottavia seikkoja. Nykyiset pelit, etenkin 3D-räiskinnät, on suunniteltu ja ohjelmoitu modulaarisesti kuvan 3 mukaisesti. 3D pelit kilpailevat keskenään enimmäkseen grafiikkamoottorien saralla ja monet niistä ovat sen johdosta julkaisseet lähdekoodin kaikista muista moduuleista.



Kuva 3: Pelien ohjelmakoodin modulaarinen rakenne(Lewis & Jacobson, 2002)

Edellisissä osioissa on mainittu Quake ja Unreal Tournament otollisina tieteellisen modauksen kohteina. Nämä pelit ovat varsin samanlaisia ja niistä on helposti löydettävissä yhtäläisiä piirteitä. Edellämainittujen vapaan lähdekoodin ja aikalaisekseen voimakkaan grafiikkamoottorin lisäksi molempiin on olemassa kenttä- ja skineditoreita<sup>1</sup>, jotka mahdollistavat uuden sisällön helpon luonnin. Lisäksi kumpikin peli on jonkinasteinen keinotodellisuusmaailma.

Voisi kuvitella, että pelejä modaamalla saisi aikaan varsin samanlaisia pelityyppisiä sovelluksia. Tässä työssä käsitellyt sovellukset ovat olleet kuitenkin selvästi erilaisia. Tähän mennessä erot on selitetty projektien täysin erilaisilla tavoitteilla. Mutta sen lisäksi käsitellyt modit ovat teknisesti erilaisia. Pelien eri moduuleita ja ominaisuuksia muokkaamalla saavutetaan erilaisia mahdollisuusavaruuksia.

### 3.1 Syöte modauksen kohteena

Syötettä muokkaamalla tähdätään vaihtoehtoisen ohjauksen implementointiin. Sinänsä jo vaihtoehtoisten syöttölaitteiden testaaminen on validi tutkimusalue, joten sellaisenaankin syötteen muokkaaminen tarjoaisi mielenkiintoisia mahdollisuuksia.

Käsitellyissä esimerkeissä syötteen muokkaaminen on kuitenkin ollut enemmänkin keino kuin tarkoitus. ARQuake ja CaveUT ovat molemmat lähteneet liikkeelle pyrkimyksestä luoda autenttinen kokemus toisesta todellisuudesta ja sen johdosta molempien syöte pyrkii kuvaamaan ihmisen normaaleja tapoja tehdä pelissä esiintyviä verbejä (liikkua, ampua, kääntyä, katsella,...). ARQuake pääsee tässä päämäärässä tietenkin pidemmälle tilarajoitteiden puuttumisesta johtuen.

Periaattellisella tasolla myös PSDoom modifioi syötettä, vaikka sinänsä hirviöiden ominaisuudet tai sijaintipaikat eivät olekaan normaalikäyttäjän ulottuvilla. PSDoom

<sup>1</sup> skinieditorilla voidaan muuttaa pelin elementtien tekstuureita. Esimerkiksi pelihahmojen ulkonäköön voi vaikuttaa skinieditorilla

onkin hyvä esimerkki siitä, että syötettä modifioidessa ei ole välttämätöntä juuttua pelin pelaajalle tarjoamaan liityntään.

Muutkin pelin hirviöitä eli tekoälyä modaavat sovellukset modaavat periaatteessa syötettä, mutta koska tekoäly on erotettavissa koodista omaksi modulikseen, käsitellään sitä erikseen.

### **3.2 Tuloste modauksen kohteena**

Tulosteen modaaminen on usein syötteen modaamista vaikeampaa, koska grafiikkamoduuli on pelintekijöiden pääomaa eikä siitä ole useinkaan lähdekoodia saatavilla.

Esimerkiksi CaveUT käyttää kuitenkin hyödykseen myös tietokoneen käyttöjärjestelmän modulaarista rakennetta joka käytännössä vaatii peliä kutsumaan grafiikka-ajureita saadakseen kuvaa ruudulle. UT tarjoaa tulosteensa OpenGL formaatissa ja sen vuoksi tuloste on kohtalaisen helppo muuntaa stereokuvaksi.

Vaikka voisi kuvitella ARQuakenkin modaavan tulostetta, on ARQuaken modifiointi enemmänkin olemassaolevien ominaisuuksien kekseliästä käyttöä. Mustat rakennukset eivät yksinkertaisesti näy puoliläpinäkyvissä virtuaalilaseissa lasien ominaisuuksien vuoksi.

Todellista ja kekseliästä modaamista tulostepuolella kokeilee SimQuake. SimQuakehän ajaa pelistä muutakin dataa tulosteena kuin pelkkää grafiikkaa. Sensoridata ekstraktoidaan pelin sisäisistä muuttujista ja tulostetaan simulaatiota käyttävän laitteen sisääntuloon.

### **3.3 Tekoäly modauksen kohteena**

Tekoäly on grafiikkamoottorin lailla jollain tavalla pelin pääomaa. Pelejähen usein arvostellaan grafiikkaominaisuuksiensa lisäksi sillä kuinka viisas tekoäly niissä on. Vaikka pelintekijät estäisivät itse tekoälyn koodin muokkaamisen, on tekoälyn skriptien muokkaaminen jo pelieditoreiden kannalta välttämätöntä. Tieteellinen modaus ei useimmiten tunnu tyytyvän pelin sisäisen tekoälyn ohjeistamiseen vaan haluaa nimenomaan kehittää tekoälyä itseään.

Koska pelejä pääsee pelaamaan, on varsin luonnollista, että pelaajan käsien kautta näppäimistön ja hiiren välityksellä singahtavat signaalit on korvattavissa tekoälyn generoimilla vastaavilla signaaleilla. GameBots on vienyt idean pelikoodin sisään liitännämoduuliksi, mutta idea on silti sama.

### **3.4 Palvelin ja verkkoyhteydet modauksen kohteena**

Verkkoyhteyksien ja palvelinsovellusten muokkaamisessa on vaikea keksiä järkevää modattavaa. CaveUT:ssa verkkoyhteyksiä ja palvelinta ei ole varsinaisesti modattu vaan

niitä on käytetty viisaasti toisin kuin ne on tarkoitettu. CaveUT käyttää verkkoratkaisua rederöinnin jakamisessa useammalle koneelle.

GameBots taas tarjoaa socket-pohjaisen liittymän ulkopuoliselle tekoälylle ollen siten periaatteessa verkkoyhteyden modaus.

### **3.5 Fysiikka, ulkonäkö ja kentät modauksen kohteena**

Pelaajat jotka modaavat pelejä omaksi ja kavereidensa iloksi modaavat nimenomaan näitä ominaisuuksia. Muutkin pelit kuin vain UT ja Quake tarjoavat usein työkaluja ulkoasun ja kenttien luomiseen. Vaikka kyseisten ominaisuuksien modaaminen onkin periaatteessa triviaalia näissä peleissä eikä vaadi mitään innovointia, sanottakoon että ilman näiden modifiointimahdollisuuksia ja työkaluja ei PSdoomia lukuunottamatta monikaan tieteellinen modaus olisi mielekäs.

Kohdissa 3.1-3.4 mainituissa tapauksissa modauksen tuloksena saadaan pelimäinen tuotos joka usein aiheuttaa rajoituksia ja vaatimuksia pelimaailmalle ja ilmapiirille. ARQuaken mallinnettu maailma pitää olla tarkalleen kuin oikea maailma ja hirviöt tavallista kirkaampia jotta ne näkyisivät päivänvalossa. SimQuaken simulointia ei varmaankaan olisi mielekästä suorittaa avaruusaluksen raunioissa, jossa möröt hyökkäilevät niskaan ja niin edelleen. Fysiikan, ulkonäön ja kenttien modaus luo kuorrutuksen ja edellytykset vaikeamman koodipohjaisen modauksen ympärille.

Etenkin taiteelliselta puolella näitä ulkoisia seikkoja modaamalla on saavutettu tuloksia, sillä modaamisen helppoudesta johtuen voidaan keskittyä asioiden kokeilemiseen eikä teknisiin seikkoihin Esimerkiksi Fuchs ja Eckerman ovat tutkineet mm. ihmisten identiteettiä modifioimalla Quaken pelikenttiä, aseita ja ulkoasua (Fuchs, 2003).

## **4 ONGELMAT JA MAHDOLLISUUDET**

Pelien modaamista hankaloittavia tekijöitä ovat tietysti edellisessä luvussa mainittujen modausta edistävien seikkojen puuttuminen. Esimerkiksi läheskään kaikkien pelien lähdekoodi ei ole vapaasti saatavilla. Kenttä- ja skineditoreita on kyllä jo tarjolla lähes jokaisessa pelissä, mutta kuten sanottua, niiden tarjoamat mahdollisuudet ovat rajalliset.

Peliteollisuudella on uskottavuusongelma (Della Rocca, 2003, s. 55-56). Yhteiskunnallisesti ja taiteellisesti pelit nähdään pikkupoikien leikkinä ja kansan moraalin turmelijana. Tästä johtuen pelien modaamisen valitseminen tieteellisen projektin alustaksi ei ole uskottavuuden kannalta helppoa. Voidaan ajatella, ettei peliin pohjautuva projekti ole mitään oikeaa tiedettä; onhan kyseessä vain peli.

Vaikka pelien modaamisella tieteellisessä tarkoituksessa on suuriakin esteitä, on tässä työssä käsitelty useita esimerkkejä joissa modaamalla pelejä on päästy kohtalaisen pienellä vaivalla kohtalaisen näkyviin ja näyttäviin tuloksiin.

Peleillä on varmasti vielä paljon tarjottavaa. Yksinkertaisesti tässä esitettyjä esimerkkejä yhdistelemällä voitaisiin jo lähteä eteenpäin. Esimerkiksi yhdistämällä PSDoom ja Soar, kyberavaruus ja tekoäly saatettaisiin päästä todelliseen ylläpitäjän

työkaluun. Kun PSDoomin hirviöt voivat yrittää yhteistyötä keskenään ja kenties jopa muita ihmismäisiä selviytymiskeinoja ja kun tekoälylle annetaan todellinen tarkoitus, tarkoitus hoitaa itselleen uskottua prosessia, saattaa tekoälytutkimus saavuttaa uusia ulottuvuuksia.

Edellä mainittu ongelma pelien uskottavuudesta voidaan toisaalta nähdä päinvastaisesta suunnasta. Mielestäni pelintekijöiden tulisi tarjota tiedemaailmalle yhä suurempia mahdollisuuksia modaamiseen. Kunhan pelejä modaamalla saavutetaan riittävän käytännöllisiä ja mullistavia tuloksia on pelejä modaavan tieteen uskottavuuden hyväksymisen kautta myös itse pelien uskottavuusongelma ratkaistu. Ainakin tällainen suuntaus antaisi peleille selkeästi positiivisen vastaanoton tiedemaailmasta.

Vaikka tässä työssä läpikäydyt pelit ovat yllättävänkin erilaisia, ovat ne kuitenkin periaatteellisella tasolla kaikki saman teeman, virtuaalimaailman, sovellutuksia. Kolmiulotteiset ensimmäisen persoonan näkökulmasta kuvatut räiskintäpelit venyvät vain kolmiulotteisiin ensimmäisen persoonan näkökulmasta kuvattuihin sovelluksiin, joissa interaktiomahdollisuudet ovat rajalliset. Jos pelien ohjelmakoodit alkaisivat vapautua muistakin genreistä saattaisimme nähdä aivan uudenlaisia sovelluksia. Toivoisin tulevaisuudessa näkeväni vaikkapa reaaliaikastrategian päälle rakennettun populaationmallinnussovelluksen tai liikennetutkimusta suorittavan autopelin. Mahdollisuudet ovat rajattomat.

## KIRJALLISUUSLUETTELO

- Bezold, B.; Diekman, N. 2003. Die 77 bedeutendsten PC-Spiele aller Zeiten. *PC Action*. Fürth, Saksa, 9/2003. s. 34–41.
- Bioware. 2003. Neverwinter Nights community site. <http://nwn.bioware.com>.
- Bylund, M.; Espinoza, F. 2002. Testing and demonstrating context-aware services with Quake III Arena. *Communications of the ACM*. Vol. 45. No. 1. s.46-48. <http://portal.acm.org/citation.cfm?id=502269.502294&dl=GUIDE&dl=ACM&CFID=12076700&CFTOKEN=96065109>
- Chao, D. 2001. Doom as an Interface for Process Management. *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*. Seattle, Washington, United States. ACM Press, New York, NY, USA, s. 152–157. <http://portal.acm.org/citation.cfm?id=365078&coll=GUIDE&dl=ACM&CFID=12076700&CFTOKEN=96065109>
- Dose, tietokonepelikauppa, [www.dose.fi](http://www.dose.fi)
- Della Rocca, J. 2003. Regulation Is Everyone's Business. *Game Developer*. Vol.10. No. 10. s.55-56
- Eckerman, S.;Fuchs, M. 2002-2003. Electronic Arts. <http://www.t0.or.at/~fuchs-eckermann/>
- Fuchs, M. 2003. Mirrors. <http://www.t0.or.at/~fuchs-eckermann/fluID/fluID.html>
- Gibson, W. (orig. 1984. Neuromancer.) 1991. Neurovelho. WSOY.
- Jacobson, J.; Hwang, Z. 2002. Unreal tournament for immersive interactive theater. *Communications of the ACM*. Vol. 45. No. 1. s.39-42.

<http://portal.acm.org/citation.cfm?id=502269.502292&dl=GUIDE&dl=ACM&CFID=12076700&CFTOKEN=96065109>

Kaminka, G. A.; Veloso, M. M.; Schaffer, S.; Sollitto, C.; Adobbati, R.; Marshall, A. N.; Scholer, A.; Tejada, S. 2002. GameBots: a flexible test bed for multiagent team research. *Communications of the ACM*. Vol. 45. No. 1. s.43-45.

<http://portal.acm.org/citation.cfm?id=502269.502293&dl=GUIDE&dl=ACM&CFID=12076700&CFTOKEN=96065109>

Laird, J. E. 2002. Research in human-level AI using computer games. *Communications of the ACM*. Vol. 45. No. 1. s.32-35.

<http://portal.acm.org/citation.cfm?id=502269.502290&dl=GUIDE&dl=ACM&CFID=12076700&CFTOKEN=96065109>

Lewis, M.; Jacobson, J. 2002. Introduction. *Communications of the ACM*. Vol. 45. No. 1. s.27-31.

<http://portal.acm.org/citation.cfm?id=502269.502288&dl=GUIDE&dl=ACM&CFID=12076700&CFTOKEN=96065109>

Piekarski, W.; Thomas, B. 2002. ARQuake: the outdoor augmented reality gaming system. *Communications of the ACM*. Vol. 45. No. 1. s.36-38.

<http://portal.acm.org/citation.cfm?id=502269.502291&dl=GUIDE&dl=ACM&CFID=12076700&CFTOKEN=96065109>

Thomas, B.; Close, B.; Piekarski, W. 2002. Interactive outdoor augmented reality collaboration system. University of South Australia.

<http://wearables.unisa.edu.au/arquake/index.html>

Tietoliikenneohjelmistojen ja Multimedian Laboratorio; Tieteen tietotekniikan keskus CSC. Eve. <http://eve.hut.fi/>

Vinge, V. 1981. True Names. *Binary Star #5*. Dell, New York